Von der Karteikarte zum Datensatz

Grundzüge der Datenspeicherung im Bibliothekswesen

Michael Horvath

Grundzüge der Datenspeicherung im Bibliothekswesen

- Datenspeicherung: Informatische Grundlagen
  - Daten, Informationen, Wissen
  - Analoge, digitale, binäre Daten
  - Datentypen, Boolesche Funktionen
  - Karteikarte - Kartei, Datensatz - Datei
  - Datenformate, Dateiformate

- Relationale Datenbanken: Modellierung & Zugriff
  - Datenbank-Architektur, Datenbank-Modelle
  - Entitäten, Attribute, Relationen
  - ERM: Entity Relationship Model
  - Schlüsselfelder, Datenintegrität
  - SQL: Structured Query Language

Grundzüge der Datenspeicherung im Bibliothekswesen

- Datenspeicherung: Informatische Grundlagen
  - Daten, Informationen, Wissen
  - Analoge, digitale, binäre Daten
  - Datentypen, Boolesche Funktionen
  - Karteikarte - Kartei, Datensatz - Datei
  - Datenformate, Dateiformate

- Relationale Datenbanken: Modellierung & Zugriff
  - Datenbank-Architektur, Datenbank-Modelle
  - Entitäten, Attribute, Relationen
  - ERM: Entity Relationship Model
  - Schlüsselfelder, Datenintegrität
  - SQL: Structured Query Language

Datenspeicherung: Informatische Grundlagen

- Was sind Daten?

- **Datum**: lat. dare = geben, ausfertigen, schreiben.

- „Die Informatik und Datenverarbeitung (EDV) benutzen Daten als (maschinen-)lesbare und -bearbeitbare **Repräsentation von Information**. Die Information wird dazu in Zeichen (ketten) kodiert, deren Aufbau strengen Regeln folgt, der so genannten Syntax...Die Semiotik definiert Daten als **potenzielle Information**.“
  Aus: http://de.wikipedia.org/wiki/Daten.

- „Man unterscheidet:
  * **strukturierte** Daten (zum Beispiel Datenbanken, XML)
  * **unstrukturierte** Daten (beispielsweise Dokumente)
  Während sich strukturierte Daten relativ einfach maschinell weiterverarbeiten lassen, ist dies bei unstrukturierten Daten nur schwer beziehungsweise ungenau möglich.“ Aus: s. o.

  → Portal zu Bibliothek - Information - Dokumentation: **http://de.wikipedia.org/wiki/Portal_BID**

Datenspeicherung: Informatische Grundlagen

- Was sind Informationen?

- **Information**: lat. in-formare = Gestalt geben, bilden.

- Nachrichtentechnik: „...Information umfasst eine **Nachricht** zusammen mit ihrer **Bedeutung** für den Empfänger. Diese Bedeutung kann darin bestehen, dass ein Mensch der Nachricht einen **Sinn** gibt, oder die Bedeutung kann indirekt aus der Art der weiteren **Verarbeitung** der Nachricht geschlossen werden..." Aus: Schülerduden „Die Informatik", 1986.

- „Information...ist potenziell oder aktuell vorhandenes, nutzbares oder genutztes **Wissen**. Wesentlich für die Information ist die **Wiedererkennbarkeit** sowie der **Neuigkeitsgehalt** anhand eines bestimmten Musters von Materie und/oder Energieformen in Raum und/oder Zeit..." Aus: http://de.wikipedia.org/wiki/Information.

Datenspeicherung: Informatische Grundlagen

- Was ist Wissen?

- **Wissen**: idg. Wurzel, ahd. wischan = gesehen haben.

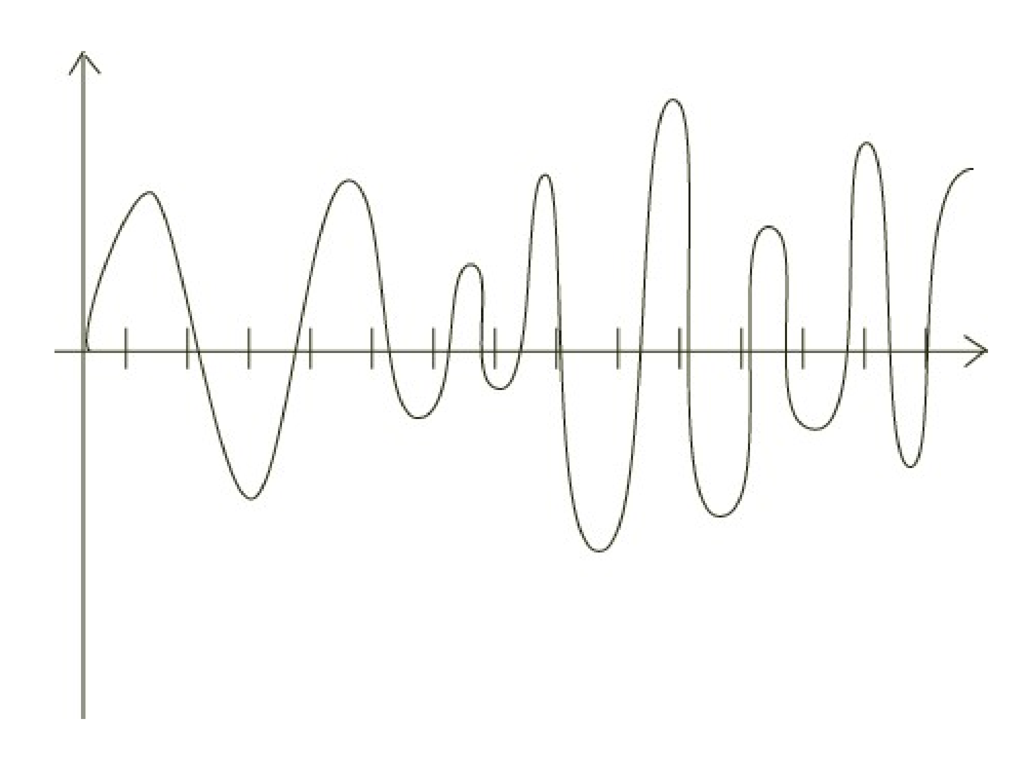
- „Wissen...bezeichnet die Gesamtheit aller **organisierten Informationen** mitsamt ihrer wechselseitigen **Zusammen-hänge**, auf deren Grundlage ein (vernunftbegabtes) System **handeln** kann. Das Wissen erlaubt es einem solchen System - vor seinem Wissenshorizont und mit der Zielstellung der Selbsterhaltung - sinnvoll und bewusst auf Reize zu reagieren." Aus: http://de.wikipedia.org/wiki/Wissen.

- „Das Wissen erfordert eine aussagekräftige **Erklärung** bzw. **Begründung**. Grundlegendes Kriterium ist das der **Überprüfbarkeit** bzw. **Nachvollziehbarkeit**. Als weiterer Maßstab gilt die Übereinstimmung des Wissens mit der wahrnehmbaren **Realität**." Aus: s. o.

Datenspeicherung: Informatische Grundlagen

- Was sind analoge Daten?

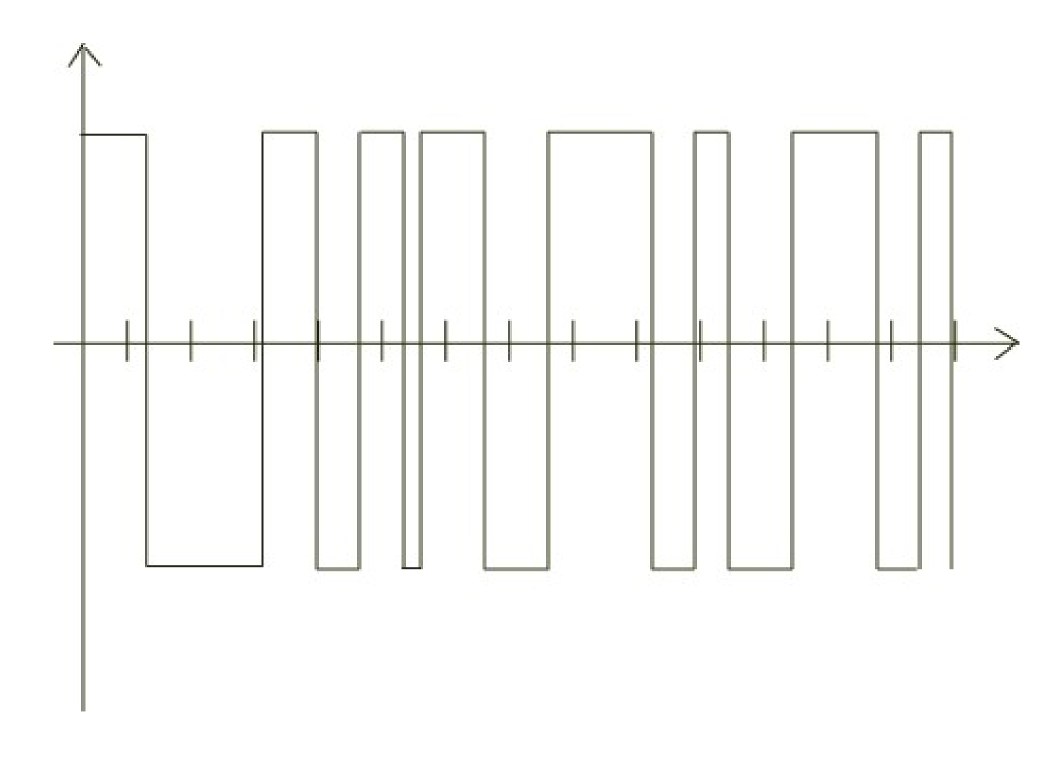
- **analog**: „**kontinuierlich**, stetig veränderbar. Eine Menge von Werten heißt *analog*, wenn sie einem **Intervall der reellen Zahlengeraden** entspricht...Analog wird in der Datenverarbeitung meist als **Gegensatz zu ‚digital'** verwendet."
  Aus: Schülerduden „Die Informatik", 1986.



Datenspeicherung: Informatische Grundlagen

- Was sind digitale Daten?

- **digital** (lat. digitus = Finger → zählen): „Eigenschaft eines Elements, nur **diskrete**, d. h. nicht stetig veränderbare Werte annehmen zu können...Ein digitales Zeichen gehört zu einem **endlichen Zeichenvorrat** mit gut unterscheidbaren Elementen...“ Aus: Schülerduden „Die Informatik“, 1986.



Datenspeicherung: Informatische Grundlagen

- Was sind binäre Daten?

- “**binär** (engl. *binary*): Ein Element heißt binär, wenn es nur zwei sich gegenseitig ausschließende Zustände annehmen

  kann. Diese Zustände werden oft mit 0 und 1...bezeichnet...“
  Aus: Schülerduden “Die Informatik”, 1986.

- Bsp. **Binäres Zählen**
  - 0 → 000
  - 1 → 001
  - 2 → 010
  - 3 → 011
  - 4 → 100

- Bsp. **Binäres Addieren**
  - 2 → 010
  - + 3 → + 011
  - = 5 → = 101

Datenspeicherung: Informatische Grundlagen

- Datenmenge: Was sind Bits und Bytes?

- „**Bit** (engl. **bi**nary digi**t**): Bezeichnung für die kleinste Darstellungseinheit für Daten in binärer Zahlendarstellung. Ein Bit kann die Werte binär Null und binär Eins annehmen.“
  Aus: Schülerduden “Die Informatik”, 1986.

- **Byte** (engl. **by** eigh**t**): „Eine Folge von 8 Bit wird als 1 Byte bezeichnet. Mit einem Byte lassen sich 256 verschiedene Objekte kennzeichnen und bei Verabredung der Codierung bedeutet die Übersendung von einer bestimmten Kombination von Null und Eins dann die Information, dass dieses dadurch bestimmte Objekt (aus einer Menge von höchstens 256 Objekten) vorgelegen hat.“
  Aus: „Wie funktioniert das? Der Computer“, 1990.

- **Kilobyte** KB 2^{10} B = 1.024 B
- **Megabyte** MB 2^{20} B = 1.048.576 B
- **Gigabyte** GB 2^{30} B = 1.073.741.824 B
- **Terabyte** TB 2^{40} B = 1.099.511.627.776 B

Datenspeicherung: Informatische Grundlagen

- Was sind Datentypen?

- „Unter einem Datentyp versteht man die Zusammenfassung von **Wertebereichen** und **Operationen** zu einer Einheit.“
Aus: Schülerduden “Die Informatik”, 1986.

- Bsp. **integer** Ganze Zahlen + - * div mod
- Bsp. **real** Reelle Zahlen + - * / Mathem. Funktionen
- Bsp. **char** Zeichenmenge ord(Zeichen) char(Zahl)
- Bsp. **boolean** true false Log. Funktionen AND OR NOT ...
- **Komplexe Datentypen** entstehen durch **Konstruktion** aus einfachen Datentypen. Bsp. **array of integer**, **string**, **record**.

Datenspeicherung: Informatische Grundlagen

- Was sind Boolesche (logische) Funktionen?

- **Boolesche Algebra**: Zweiwertiges formales System, das 1854 vom englischen Logiker George **Boole** (1815 - 1864) entwickelt wurde.

- 1stellige Funktion **NOT** ($\neg$) : A sei eine Aussage. Die Aussage NOT A ist genau dann wahr, wenn A falsch ist. $\rightarrow$ **Negation**.

- 2stellige Funktion **AND** ($\wedge$) : A , B seien Aussagen. Die Aussage A AND B ist genau dann wahr, wenn **sowohl** A **als auch** B wahr ist. $\rightarrow$ **Konjunktion**, **Durchschnittsmenge**, **Serienschaltung**.

- 2stellige Funktion **OR** ($\vee$) : A , B seien Aussagen. Die Aussage A OR B ist genau dann wahr, wenn **mindestens eine** der beiden Aussagen A , B wahr ist. $\rightarrow$ **Disjunktion**, **Vereinigungsmenge**, **Parallelschaltung**.

-

A	NOT A
f	w
w	f

A	B	A AND B
f	f	f
f	w	f
w	f	f
w	w	w

A	B	A OR B
f	f	f
f	w	w
w	f	w
w	w	w

Datenspeicherung: Informatische Grundlagen

- Daten vs. Metadaten

- **meta** : griech. inmitten, hinter (örtlich), nach (zeitlich).

- **Metadaten** werden **im nachhinein** aus Daten gewonnen. Sie sind **Daten über Daten**.

- Im Bibliothekswesen entstehen Metadaten durch
  - **Formale Erfassung** → vgl. Nominalkatalog
  - **Inhaltliche Erschließung** → vgl. Schlagwortkatalog

- **Informatische Perspektive** auf Datenspeicherung im Bibliothekswesen:

- Phase 1: Daten **analog** Metadaten **analog**
- Phase 2: Daten **analog** Metadaten **digital**
- Phase 3: Daten **digital** Metadaten **digital**

Datenspeicherung: Informatische Grundlagen

- Exkurs: KatZoom-Kataloge
  → http://katzoom.onb.ac.at/katzoom/

- **KatZoom**: Online-Zugriff auf **digitalisierte Zettelkataloge**, Entwicklung der ÖNB, Service auch für andere österreichische Bibliotheken, insgesamt **> 12 Mio. Zettel online** verfügbar.

- Besonderheit: **Zoom-Funktion**
  - Alphabetische Ordnungsfolge der Zettelkataloge wird beibehalten.
  - Ausgewählter Suchbereich (Buchstabe) wird in 4 Teile geteilt.
  - Kurzanzeige mit erstem Zettel aus jedem der 4 Teile und letztem Zettel des Suchbereichs ermöglicht immer nähere Eingrenzung.

- Bsp. Wichtige KatZoom-Kataloge an der ÖNB

  - Nominalkatalog 1501 - 1929 `1.114.000 Zettel`
  - Nominalkatalog 1930 - 1992 `1.424.000 Zettel`

  - Schlagwortkatalog 1501 - 1929 `1.356.000 Zettel`
  - Schlagwortkatalog 1930 - 1992 `1.690.000 Zettel`

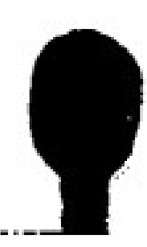
Datenspeicherung: Informatische Grundlagen

- Bsp. Zettel aus „Nominalkatalog 1930 - 1992"
  Aus: http://katzoom.onb.ac.at/katzoom/.

62412 B

S c h r a f f , Franz

Praktische erprobte Kalkulation für Fleisch und Wurst.2.verb.Aufl.-München:Pflaum[1955].
21 S.8°



Datenspeicherung: Informatische Grundlagen

- Bsp. Zettel aus „Schlagwortkatalog 1930 - 1992"
  Aus: http://katzoom.onb.ac.at/katzoom/.

Wurst: Kalkulation 62412 B

S c h r a f f , Franz:Praktisch erprobte Kalkulation für Fleisch und Wurst.
2.verb.Aufl.-München:Pflaum
[1955].21 S.8°



Datenspeicherung: Informatische Grundlagen

- Was ist eine Kartei?

- „Eine Kartei ist eine **geordnete Sammlung von Daten** auf kleinformatigen Karten, wobei jede der Karten für eine **Informationseinheit** steht und damit ein **Objekt der Wirklichkeit** repräsentiert."
  Aus: http://de.wikipedia.org/wiki/Kartei.

- „Zum effizienten Benutzen einer Kartei ist die **korrekte Anordnung der Karten** unbedingt erforderlich; andernfalls lassen sich Daten nicht finden." Aus: s. o.

- „Eine Kartei kann auf eine **Datenbank** elektronisch abgebildet werden. Im Gegensatz zu einer Datenbank besteht eine **Kartei** allerdings meist nur aus einer einzigen **Ansammlung strukturgleicher Karteikarten**." Aus: s. o.

- „Die elektronische Entsprechung einer **Kartei** ist demnach die **Tabelle** einer Datenbank. Eine **Karteikarte** entspricht dann einer **Zeile** in dieser Tabelle." Aus: s. o.

Datenspeicherung: Informatische Grundlagen

- Lochkarte: Vorläufer

- Ursprung: **Spieldosen** und ähnliche Apparate, in denen sich drehende Walze oder Scheibe mit Stiften oder Löchern **Wiedergabe von Musikstücken** ermöglicht.

- **Lochkartenähnliche Systeme** etwa ab Mitte des 18. Jh. enthalten **Anweisungsfolgen für wiederkehrende Abläufe**. Bsp. Webstühle, Drehorgeln.

- **Randlochkarte**: „Verschiedene **Suchkriterien** (zum Beispiel im Bibliothekswesen) wurden mit **Löchern** oder **Schlitzen** an allen 4 Rändern der Karte codiert. Man konnte mit einer langen **Nadel** in die Position eines Suchkriteriums stechen und die ausgewählten Karten aus dem **Stapel** herausziehen. "
  Aus: http://de.wikipedia.org/wiki/Lochkarte.

Datenspeicherung: Informatische Grundlagen

- Lochkarte: Maschinenlesbare Karteikarte

- „... Lochkartenformat geht auf die US-amerikanische **Volkszählung 1890** zurück, zu der **Herman Hollerith** ein auf Lochkarten basierendes Verfahren einschließlich der zugehörigen **Stanz- und Auswertemaschinen** entwickelte."
  Aus: http://de.wikipedia.org/wiki/Lochkarte.

- „Eine **Hollerith-Lochkarte** ist ein rechteckiges, etwa **18.7 cm × 8.3 cm** großes Stück 0.17 mm dünner Karton, in das in vorgegebene Positionen **spaltenweise** Löcher gestanzt werden um eine Folge von Zeichen (heute würde man sagen: **eine Zeile Text**) zu codieren." Aus: s. o.

- **Lesen**: **mechanisch** (Stifte), dann **optisch** (Lichtschranken).

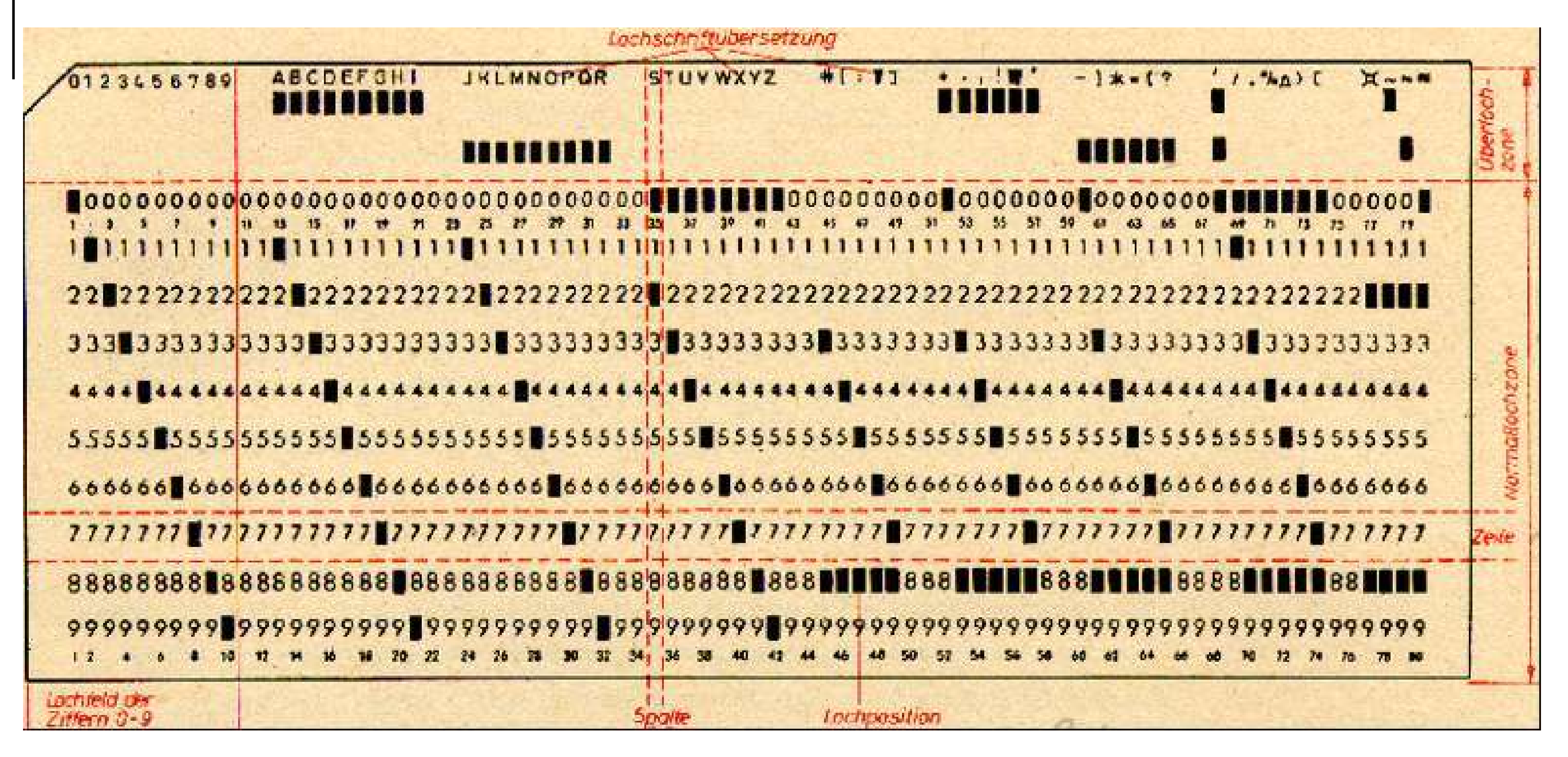
- **Schreiben**: **Lochkartenstanzer** mit Tastatur.

- 3 **Lochkartenstapel** (**batch** → batch job, batch file):
  für **Eingabedaten**, für **Programm**, für **Ausgabedaten**.

Datenspeicherung: Informatische Grundlagen

- Lochkarte mit 80 Zeichen: IBM-Patent von 1928

- „Jedes Zeichen wird durch eine bestimmte **Lochkombination** dargestellt, wobei **Ziffern** durch ein Loch, **Buchstaben** und **Sonderzeichen** durch zwei oder drei Löcher codiert werden.“
Aus: Schülerduden “Die Informatik”, 1986. → Zonenteil / Ziffernteil.



Datenspeicherung: Informatische Grundlagen

- Was sind Datenformate? (1)

- **Datenformat**, **Zeichensatz**, **Code**: „Abbildungsvorschrift, die jedem Zeichen eines Zeichenvorrats (*Urbildmenge*) eindeutig ein Zeichen oder eine Zeichenfolge aus einem möglicherweise anderen Zeichenvorrat (*Bildmenge*) zuordnet.“
  Aus: Schülerduden “Die Informatik”, 1986.

- **Morse-Code**: Zeichenfolgen verschiedener Länge, daher nicht umkehrbar eindeutig. Bsp. e = **.** z = **- - . .**
- **Telegraphen-Code**: **5 Bit** → Codierung von $2^5 = 32$ Zeichen.

- **Hollerith-Code**: **12 Bit**. → Zeilen einer Lochkarte mit/ohne Lochung. 10 Ziffern, 26 Buchstaben, 28 Sonderzeichen.
- **BCD-Code**: Binary Coded Decimals. **4 Bit** (1 Halbbyte). → Binäre Darstellung der dezimalen Ziffern 0 - 9.
- **EBCDIC**: Extended Binary Coded Decimals Interchange Code. **8 Bit** (1 Byte): Linkes Halbbyte Zonenteil, rechtes Ziffernteil. Verwendung im Bereich der **Großrechner** bis heute.

- Was sind Datenformate? (2)

- **ASCII**: American Standard Code for Information Interchange. Standard seit 1968. **7 Bit + 1 Prüfbit**: Zeichen von 0 bis 127 fortlaufend numeriert. Bsp. „A“ hat Code 65.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
16	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
32	SP	!	»	#	$	%	&	'	(	)	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

- **ANSI-Code**: American National Standards Institute. **8 Bit**. Verwendung in ersten Versionen von MS Windows.
- **ISO-8859-Codes**: Internat. Organisation for Standardization. **8 Bit**. Notwendigkeit durch Verbreitung des Internets. Bsp.:
  - ISO-8859-1 (ISO-Latin-1): ähnlich ANSI. USA und Westeuropa.
  - ISO-8859-2 (ISO-Latin-2): Slawische Sprachen.
  - ISO-8859-3 (ISO-Latin-3): Esperanto, Maltesisch, Türkisch.
  - ISO-8859-4 (ISO-Latin-4): Baltische Sprachen (Estnisch, Lettisch, Litauisch).

Datenspeicherung: Informatische Grundlagen

- Was sind Datenformate? (3)

- **UNICODE**: **16 Bit** (2 Byte) für 2^{16} = 65.536 Zeichen, um Problem der Darstellung von Zeichen endgültig zu lösen. Enthält Hiragana, Katakana, CJK Unified Ideographs (für chinesich, japanisch, koreanisch) etc. Erste 256 Zeichen entsprechen **ISO-Latin-1**. Windows NT 4.0 1996 erstes Betriebssystem, das 16-Bit-UNICODE unterstützt.

- **UTF-8** (UNICODE Transformation Format 8 Bit) derzeit erfolg-reichste UNICODE-Umsetzung, Bytefolgen variabler Länge von **1 bis 4 Byte**. 1-Byte-Variante entspricht ASCII-Code. → XML.

- Bsp.: Zeichen aus Schriftart **Arial Unicode MS** unter MS Word

armenisch	tamilisch	japanisch
	ஞுர்ருழொ	あのれシキホ
		平雪迎骨水直

Datenspeicherung: Informatische Grundlagen

- Was ist eine Datei?

- **Datei** (von **Dat**en + Kart**ei**, engl. **file**): „Bezeichnung für eine nach bestimmten Gesichtspunkten zusammengestellte **Menge von Daten**. Eine Datei besteht aus einer Folge von gleichartig aufgebauten **Datensätzen**; jeder Datensatz setzt sich im allgemeinen aus mehreren unterschiedlichen Komponenten, den **Feldern**, zusammen." Aus: Schülerduden "Die Informatik", 1986.

- Anfangs **sequenzielle Dateiorganisation** auf Magnetband: Datensätze können nur in Reihenfolge der Eingabe gespeichert und wieder abgerufen werden. → „Virtueller Lochkartenstapel".

  Bsp. Datei mit 3 Datensätzen fixer Länge zu je 130 Zeichen

```
Nr.Autor                Titel                    Jahr
|  |5   10   15   20|  25   30   35   40 ... |130
001Kant, I.             Kritik der reinen Ve ... 1787
002Nietzsche, F.        Jenseits von Gut und ... 1886
003Wittgenstein, L. Tractatus logico-phi ... 1921
```

Datenspeicherung: Informatische Grundlagen

- Dateiformate: Text Files vs. Binary Files

- **Textdatei**: Speicherung als **menschenlesbarer Klartext**, Bearbeitung durch **Editor**. „Wichtige Beispiele sind **XML**, **HTML**, die Satzsprache **LaTeX**, die Druckseitenbeschreibungs-sprache **PostScript** und nicht zuletzt die zahlreichen **proprietären Konfigurationsdateien** für Betriebssysteme und Anwendungen.“ Aus: http://www.galileocomputing.de.

  → Buch „Kompendium der Informationstechnik“ **gratis** downloadbar: **http://www.galileocomputing.de/openbook/kit/index.htm**

- **Binärdatei**: „Es handelt sich um Dateien mit **numerischen Inhalten**, wobei ohne Kenntnis der **Spezifikation** des jewei-ligen Dateiformats nicht erkennbar ist, ob die gespeicherten Zahlen an bestimmten Stellen für **Befehle** oder für **Nutz-daten** stehen.“ Aus: s. o.
  - Grafik-Dateiformate `.bmp .gif .jpg .tif`
  - Audio-Dateiformate `.wav .mp3`
  - Video-Dateiformate `.mov .avi .mpg`

Datenspeicherung: Informatische Grundlagen

- Bsp. Öffnen einer Binärdatei im Dateiformat .gif in einem Editor für Textdateien
-


```
______________________________________________________________________
GIF89aÜ v ÷  ŒŠŒDFDÌÎÌ$"$¬ª¬dfdìîìœšœTVTÜÞÜ424¼º¼tvtüþü  ^
–  qß Í  « °  Ü  "       o^| åç    V¢^ [ Ô– wq°|œæç     ð^;
úÔ ww ^ åàõý w Sÿœ!ÿççÿwÿ  7O ;õÔ ww Ëh `çç w ¨ 3ä ;Ô  w
ÈÞ^å`'ç~ w ðpœ^Ò'ú~w  ^¸ çõ w  ÿ ÿ  ÿ  ÿ  7„æ õ w  Änq ç–
wq          Ú        ÐX )æ            ÜŒ âç      ~ H  é Àw  ^P
2ç è w ÿÿÿÿèÿÿÿÿ ÿÞlÿ`[ÿççÿww ª ‡î – q  p Òî      | çî
l^  à – q `          Äöæ æ w  \^äç÷–wwq      @    8ä aæ
È  ( ç  w   å          l  [  ç  wà8Aøa     ð ÿ^ÿú ÿw ÿ^A) õç
ww ÿTÿçòÿÿ  7d / õè ww à  èç w   þ îXK    oì ç ç w 8ÿ[aÿbÿN
ÿ  èè    dõ/èççwww lÉ oè N    (½ –ÿ Nÿ   5 ^EèÔ w    çî
o  ç  w À Üî O       0        d4 æç     £Pn]:ý÷ÔFww !ù   ,
Ü v Gÿ H° Áƒ*\È°¡Ã‡#JœH±¢Å‹3jÜÈ±ãE  xI0¤HLªÀÀ’%Cv4       ³à•&
     (( Q@H/  &é !H â\šà
M„CU0ÐàáÍ>u*“ãÖ^,@tÉ¤1OL‰´`TÃ0 ÐI²]*YUã¶ Š.\J¸ðJ„6–
øõA"}+`«"ç<82³Ã>       ,&èÓ$´W,½ÒÀšŒóZ$¶íÛ¸sëÞÍ»·ïßÀƒ
______________________________________________________________________
```

Grundzüge der Datenspeicherung im Bibliothekswesen

- Datenspeicherung: Informatische Grundlagen
  - Daten, Informationen, Wissen
  - Analoge, digitale, binäre Daten
  - Datentypen, Boolesche Funktionen
  - Karteikarte - Kartei, Datensatz - Datei
  - Datenformate, Dateiformate

- Relationale Datenbanken: Modellierung & Zugriff
  - Datenbank-Architektur, Datenbank-Modelle
  - Entitäten, Attribute, Relationen
  - ERM: Entity Relationship Model
  - Schlüsselfelder, Datenintegrität
  - SQL: Structured Query Language

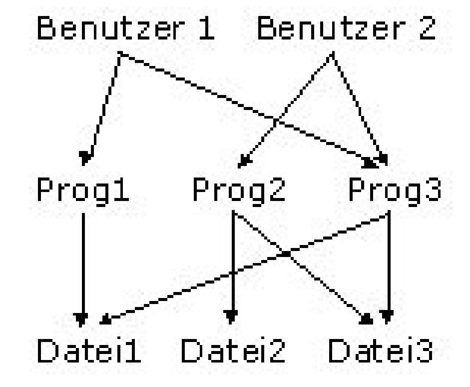
Relationale Datenbanken: Modellierung & Zugriff

- Was ist eine Datenbank?

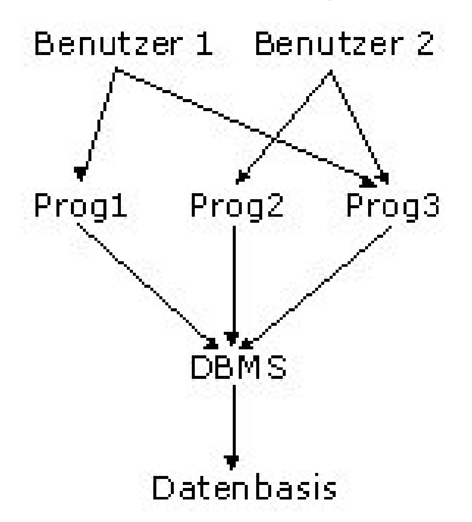
- **Datenbank**: „System zur **Beschreibung**, **Speicherung** und **Wiedergewinnung** von umfangreichen Datenmengen, die von **mehreren Anwendungsprogrammen** benutzt werden.“
  Aus: Schülerduden “Die Informatik”, 1986.

- „Es besteht aus der **Datenbasis**, in der die Daten abgelegt werden und den Verwaltungsprogrammen (*Datenbank-software*, **Datenbankmanagementsystem**)...“ Aus: s. o.

- Datenspeicherung in Dateisystem

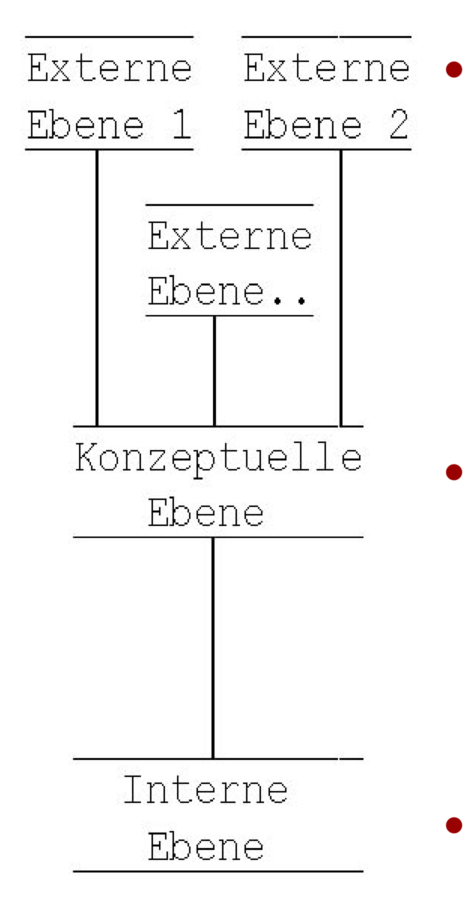


Datenspeicherung in Datenbanksystem



Relationale Datenbanken: Modellierung & Zugriff

- Datenbank-Architektur: 3-Ebenen-Modell (1975)



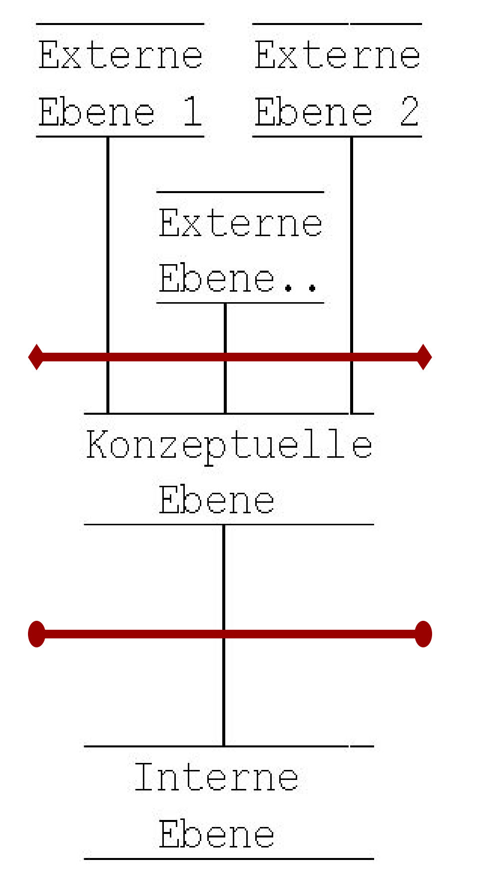
- **Externe Ebene**: Sichten (**Views**) bestimmter **Anwender** auf die DB, wobei jeder nur die **für ihn relevanten** Daten sieht.
  OPAC-Benutzer → Autor, Titel, Jahr...
  Infoschalter → zusätzlich Entlehnstatus
  Katalogisierer → alle bibliographischen Daten
  Administrator → ALLE Daten, auch über Zugriffe

- **Konzeptuelle Ebene**: **Logische** Struktur der DB, die **Daten**, deren **Eigenschaften** und **Beziehungen** zueinander beschreibt.
  → **Datensatz** „Buch" kann mit mehreren Sätzen „Exemplar" in Relation stehen, enthält 50 Zeichen langes **Datenfeld** „Autor" vom **Datentyp** „string" etc.

- **Interne Ebene**: **Physische** Struktur der DB, die Details zur **Speicherorganisation** enthält und **Leistungsfähigkeit** der DB bestimmt.
  → Auf welcher Festplatte welches Rechners liegen die Daten? Welche Mechanismen werden verwendet, um möglichst schnell darauf zugreifen zu können? ...

Relationale Datenbanken: Modellierung & Zugriff

- Datenbank-Architektur: Datenunabhängigkeit



- **Logische Datenunabhängigkeit**: Änderungen auf der **konzeptuellen** und damit auch **internen** Ebene haben (meist) keine Auswirkungen auf die externe Ebene.
  → Wird bei jedem Datensatz „Buch" ab sofort ein Datenfeld „ISBN-Nummer" mitgespeichert, müssen auf der externen Ebene nur jene Anwendungsprogramme verändert werden, die das Feld tatsächlich benutzen sollen.

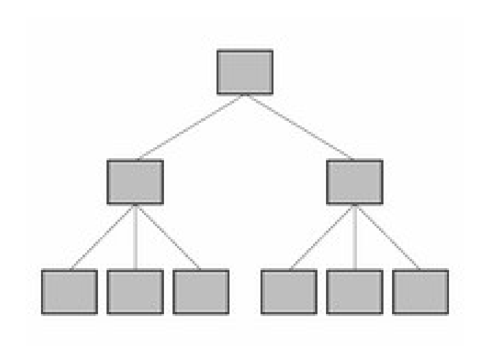
- **Physische Datenunabhängigkeit**: Änderungen auf der **internen** Ebene haben keine Auswirkungen auf die darüber liegenden Ebenen.
  → Wird auf der internen Ebene ein bestimmter Zugriffsmechanismus optimiert, muss auf der externen Ebene kein Anwendungsprogramm umgeschrieben werden.

Relationale Datenbanken: Modellierung & Zugriff

- Datenbank-Modelle: Hierarchische Datenbank

- Ältestes Modell, bereits **Ende der 50er-Jahre** für Einsatz auf **Großrechnern** entwickelt. **DL/1 von IBM** im Bereich von Banken und Versicherungen heute noch im Betrieb, war 1990 noch schnellste DB mit > 1000 Transaktionen/Sekunde!

- Verknüpfung von Datensätzen inform einer hierarchischen **Baumstruktur**: Es gibt genau eine „**Wurzel**" und jeder Datensatz hat genau einen „**Vorgänger**". → **Parent** / **Child**.



- Bsp. Abfrage einer DL/1-DB: Überstunden des Mitarbeiters Nr. 130 vom 05. 01. 1983

```
CALL CBLDLI (6, GU, PCB, WAP06,
Q-SSA-P01, Q-SSA-P05, Q-SSA-P06)

Q-SSA-P01: P01      (P01PNR  = 130)
Q-SSA-P05: P05      (P05ADAT = 8301)
Q-SSA-P06: P06      (P06ATAG = 05)
```

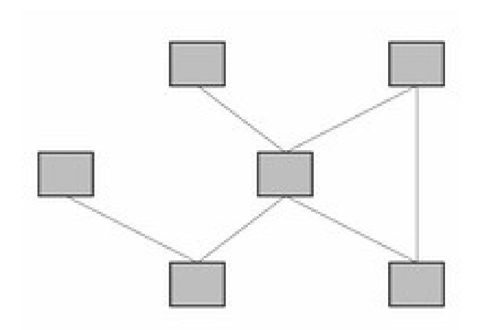
GU ... Get Unique (Direktzugriff)
SSA Segment Search Argument

Relationale Datenbanken: Modellierung & Zugriff

- Datenbank-Modelle: Netzwerk-Datenbank

- Modell wurde **1974** als **ANSI-Norm** verabschiedet, IBM stimmte dagegen, weshalb es wenig Verbreitung fand. **UDS**: Wichtigste DB für **SIEMENS**-Großrechner unter BS2000.

- Verknüpfung von Datensätzen als **Netzwerk**: Suche, auch: **Navigation**, kann **mehrere Einstiegspunkte** („Wurzeln") haben, Datensatz kann **mehrere „Vorgänger"** haben.



- Hierarchische und Netzwerkdatenbanken $\rightarrow$ **Satzorientierte Datenbanken**:
  - Entwicklung durch **Hardware-Firmen**
  - **Hohe Performance** auf Großrechnern
  - **Hoher Aufwand an Personal** für Entwicklung und Betrieb
  - **KEINE** logische bzw. physische **Datenunabhängigkeit** gegeben: Umschreiben aller Anwendungen bei konzeptuellen / internen Änderungen

Relationale Datenbanken: Modellierung & Zugriff

- Datenbank-Modelle: Relationale Datenbank

- **1970 - 1975**: IBM-Forschungsprojekt unter Edgar F. **Codd** zur Entwicklung eines Konzepts für relationale Datenbanken. Prototyp: **System/R** mit Zugriffssprache **Sequel**, einem Vorläufer von **SQL** (Structured Query Language).

- Teil der Gruppe entwickelte **IBM DB2**, anderer Teil gründete Firma **Oracle**, die **1978** erstes System auf den Markt brachte. Andere Systeme: MS **SQL Server**, MS **Access**, **mySQL** (frei).

- **Objekte**, deren **Eigenschaften** und **Beziehungen** zueinander werden im relationalen Modell durch **Tabellen** dargestellt. Verbindung von Tabellen durch **Schlüsselfelder** mit gleichem Wert. → **Mengenorientierte Datenbanken**.

P-ID	Name	Vorname	Ort
1	Müller	Hans	Oberdorf
2	Meier	Jakob	Hinterwil
3	Keiser	Josef	Unterdorf
...	...	...	...

Ort-ID	Fläche	Art	...
5689	45869	Dorf	...
4758	23864	Weiler	...
2145	86541	Stadteil	...
...	...	...	...

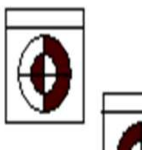
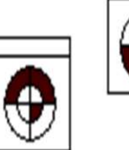
Relationale Datenbanken: Modellierung & Zugriff

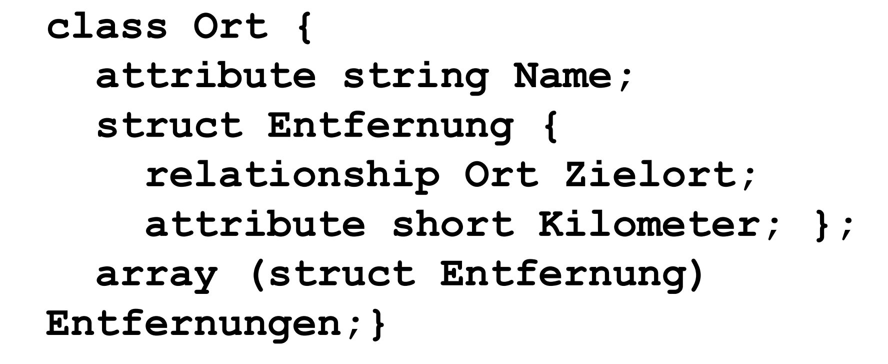
- Datenbank-Modelle: Objektorientierte Datenbank

- **Objektorientiertes Paradigma** kommt aus Programmierung. Definition von **gekapselten Objekten** mit deren **Attributen** / **Methoden**, welche untereinander **Nachrichten** austauschen können. Hierarchische Anordnung in **Klassen**, Möglichkeit der **Vererbung** von Attributen und Methoden.

- **OODBMS** (Object Oriented DB Management System) erlaubt im Gegensatz zu RDBMS frei definierbare **Schachtelung** von komplexen Strukturen. Umsetzung: **Caché** von Intersystems.
  - Bsp. Definition einer Klasse „Ort“ zum Speichern von Distanzen in OODBMS



```
class Ort {
  attribute string Name;
  struct Entfernung {
    relationship Ort Zielort;
    attribute short Kilometer; };
  array (struct Entfernung)
Entfernungen;}
```

- Datenbank-Modelle: Objektrelationale Datenbank

- **Erweiterung von RDBMS** um grundlegende objektorientierte Konzepte, ohne Komplexität von OODBMS zu erreichen. Bsp. Oracle 8i / 9i, IBM DB2 UDB, Informix Dynamic Server.

- ORDBMS erlaubt Verwendung von Objekten als **selbst definierte komplexe Datentypen** und Darstellung von **Relationen** zwischen **Mengen von Objekten**.

  Bsp. GIS - Geographisches Informationssystem: Koordinaten stehen in Relation zueinander und bilden Objekt „Position". Positionen stehen in Relation zueinander und bilden Objekt „Straße".

ID	XY	DF	ER
56		XXX	
45		YYY	
...	...	...	...

- Was sind Entitäten und Attribute?

- **Entität**, Objekt, Tupel: **Individuelles** und **identifizierbares** Exemplar von **Dingen** oder **Begriffen** der realen oder der Vorstellungswelt. Bsp. Autor „Albert Camus“, Buch „Der Fall“, Schlagwort „Existenzialismus“.

- **Attribut**, Eigenschaft: Merkmal, das einer Entität zugeordnet wird und deren **Charakterisierung**, **Klassifizierung** oder **Identifizierung** ermöglicht. Bsp. Autor „Albert Camus“ mit Attributen Geburtsjahr „1913“ und Sterbejahr „1960“.

- **Entitätstyp**, Objekttyp: **Klasse von Objekten** mit körper-licher oder begrifflicher Existenz. Bsp. Dublin-Core-Felder „Title“, „Creator“, „Subject“ etc. sind Entitätstypen.

-

Traditionelle EDV	Datenmodellierung	RDB
Datensatz	Entität	Zeile
Datenfeld	Attribut	Spalte
Datei	Entitätstyp	Tabelle

- Was sind Relationen?

- **Relation**, Beziehung: Verbindung von zwei **Entitätstypen**.
Bsp. Autor „hat verfasst" Buch. Exemplar „liegt in" Regal.
(**Jede Tabelle** einer RDB ist selbst eine Relation, daneben können Relationen **zwischen Tabellen** definiert werden.)
Relation kann auch **Attribute** aufweisen. Bsp. **Kardinalität**:

- **1 : 1 - Relation**: **Eine** Entität des ersten Typs ist entweder **mit keiner** oder **mit genau einer** Entität des zweiten Typs verbunden.
Bsp. Mann „ist verheiratet mit" Frau. (In monogamen Gesellschaften hat ein Mann entweder keine oder eine Frau.)

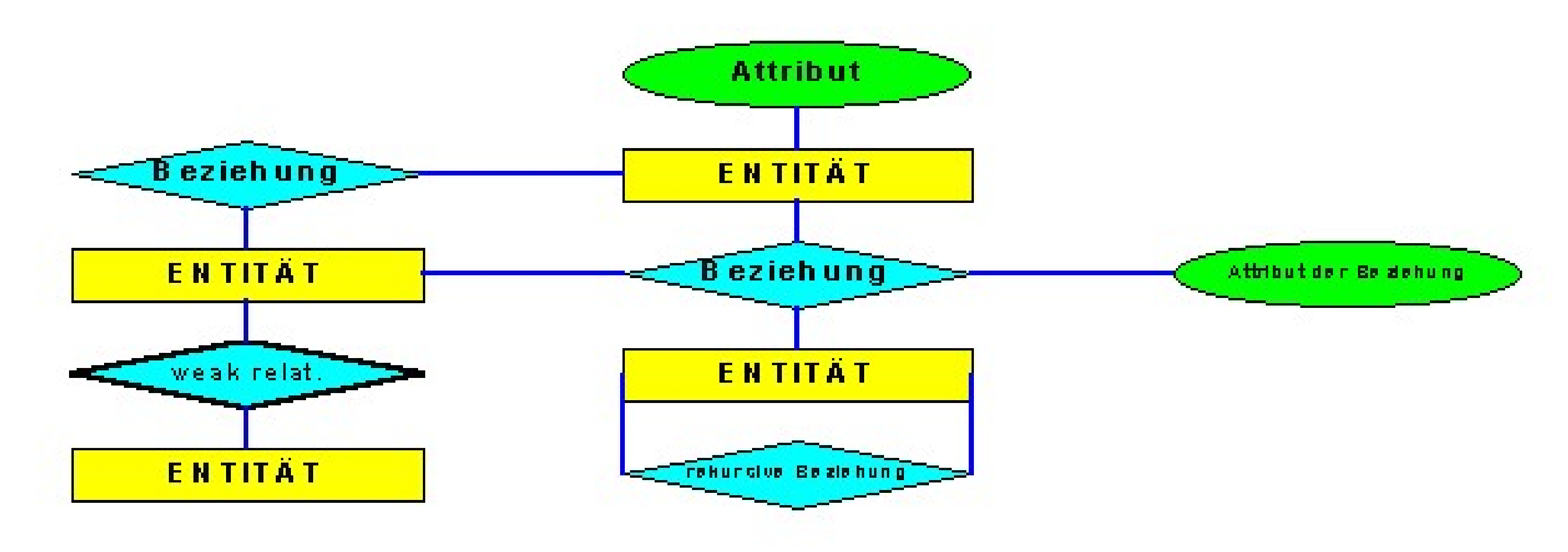
- **1 : n - Relation**: **Eine** Entität des ersten Typs ist mit **beliebig vielen** Entitäten (0, 1 oder mehr) des zweiten Typs verbunden.
Bsp. Abteilung „beschäftigt" Mitarbeiter. (Abteilung hat mehrere Mitarbeiter, keiner von ihnen kann in mehreren Abteilungen arbeiten.)

- **m : n - Relation**: **Beliebig viele** Entitäten des ersten Typs sind **mit beliebig vielen** Entitäten des zweiten Typs verbunden.
Bsp. Person „entlehnt" Buch. (Person kann mehrere Bücher entlehnen, Buch kann von mehreren Personen entlehnt werden.)

- Entity Relationship Model:
  Bedeutung der Symbole

- Entitätstyp **Rechteck**
  Attribut **Ellipse**
  Relation Linie - **Deltoid** - Linie

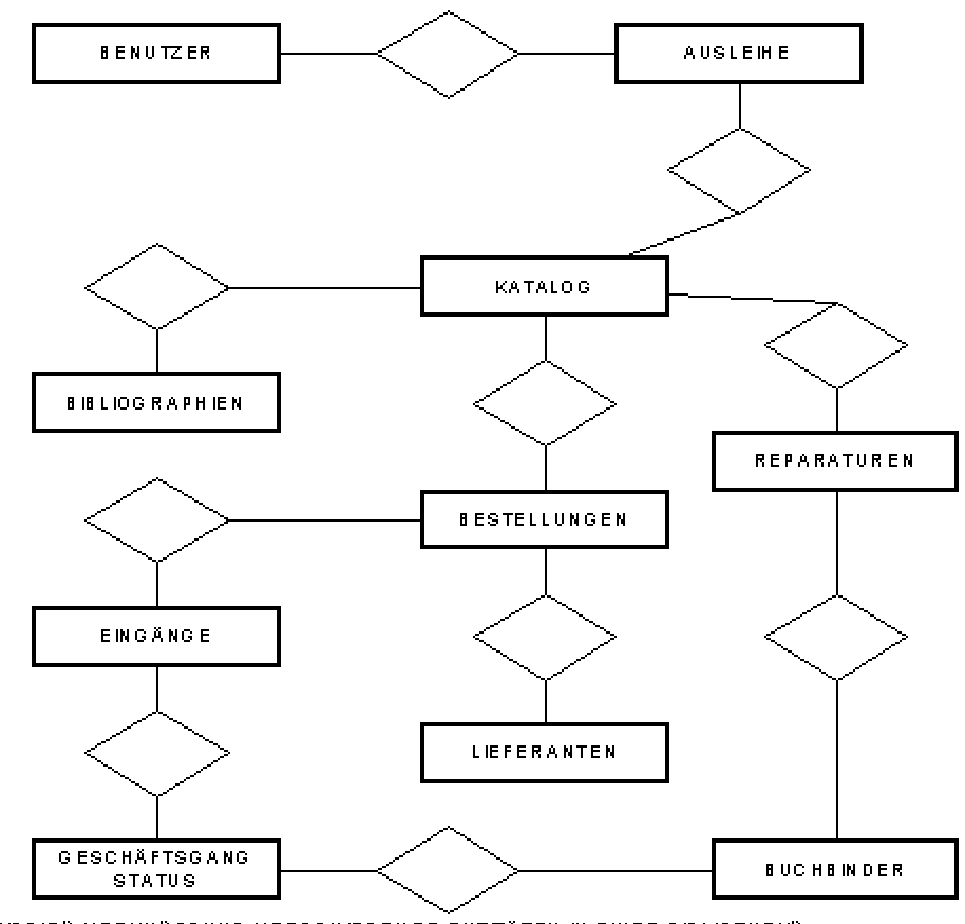
Diagramm-Symbole für Beziehungen (relationships)



dbauf06.gif

Relationale Datenbanken: Modellierung & Zugriff

- Entity Relationship Model:
  Beispiel für Grob-ERM einer Bibliothek



BEISPIEL: VERKNÜPFUNG VERSCHIEDENER ENTITÄTEN IN EINER BIBLIOTHEK

- Was sind Primärschlüssel und Fremdschlüssel?

- **Primärschlüssel**, Entitätsschlüssel, **primary key**: dient der **eindeutigen Identifizierung** einer Entität in einer RDB, kann aus einem oder mehreren Attributen bestehen.

  Bsp. ISBN eines Buches; Signatur eines Exemplars, bestehend aus ISBN und laufender Nummer; Kundennummer eines Bibliotheksbenutzers.

- **Fremdschlüssel**, **foreign key**: dient der Realisierung von **Beziehungen zwischen Tabellen** in einer RDB, ist ein **Primärschlüssel aus einer anderen Tabelle**.

  Bsp. ISBN ist Primärschlüssel in Tabelle „Buch" und Fremd-schlüssel in Tabelle „Exemplar", weil sich Exemplare immer auf ein bestimmtes Buch beziehen müssen. → 1 : n - Relation.

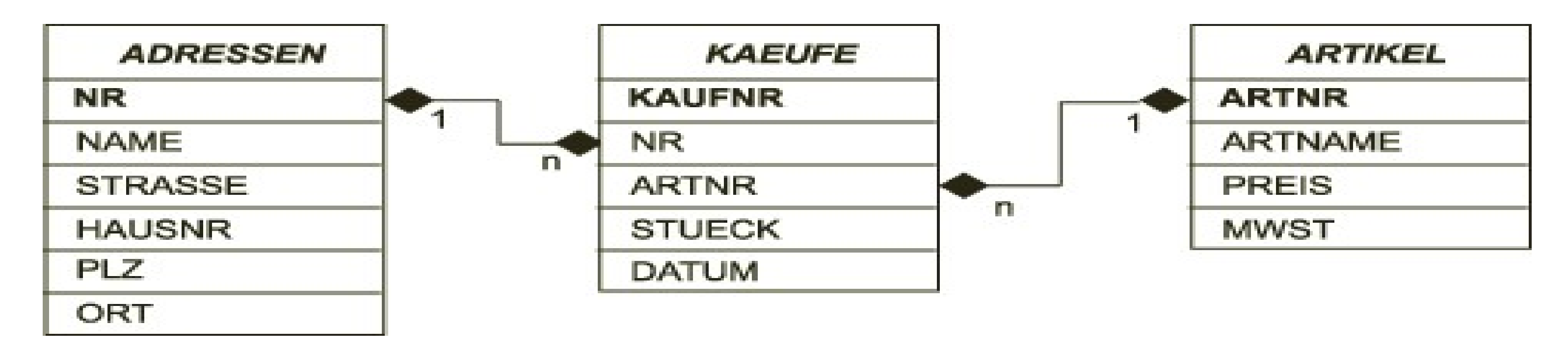
- Bsp. Tabellen mit Schlüsseln und Beziehungen

- 3 Tabellen: ADRESSEN (→ Käufer), KAEUFE, ARTIKEL.
  2 Relationen: ADRESSEN (→ Käufer) „tätigen" KAEUFE.
  ARTIKEL „werden gekauft durch" KAEUFE.

  Anm.: ADRESSEN „kaufen" ARTIKEL wäre **m : n - Relation**, die in einer RDB **nicht direkt abgebildet**, sondern nur durch **zwei 1 : n - Relationen** dargestellt werden kann. KAEUFE ist „Zwischentabelle".

- Primary K.: ADRESSEN.NR, KAEUFE.KAUFNR, ARTIKEL.ARTNR.
  Foreign K.: KAEUFE.NR, KAEUFE.ARTNR.

  Anm. Schreibweise: **<Tabellenname>.<Feldname>**. Relation immer **vom Primärschlüssel (1) zum Fremdschlüssel (n)** dargestellt.



- Datenintegrität: Prüfung durch das RDBMS

- **Entitäts-Integrität**, Objekt-Integrität: Jeder Primärschlüssel darf **nur einmal vergeben** werden, damit die dazugehörige Entität **eindeutig identifizierbar** bleibt. Jedes Attribut, das Teil eines Primärschlüssels ist, muss **stets befüllt** sein. Anm.: Einem Attribut, das momentant keinen expliziten Wert enthält, weist das System den **symbolischen NULL-Wert** zu, der **nicht 0, sondern LEER** bedeutet. Ein Attribut, das Teil eines Primärschlüssels ist, darf also **niemals NULL** sein.

- **Referenzielle Integrität**, Verweis-Integrität: Ein Fremdschlüssel darf **nicht ohne dazugehörigen Primärschlüssel** vorhanden sein. Bsp. Ein Benutzer, der noch ein Buch ausgeliehen hat, darf nicht gelöscht werden können.

- **Wertebereichs-Integrität**, Domänen-Integrität: Abgesehen vom Datentyp eines Attributs, kann dessen **Wertebereich per Definition eingeschränkt** werden. → **Constraints**. Bsp. Feld „Status“ in Tabelle „Exemplar“ darf nur Werte „P“ (am Platz), „E“ (entlehnt), „R“ (in Reparatur) oder „U“ (unbekannt) enthalten.

Relationale Datenbanken: Modellierung & Zugriff

- Was ist SQL?

- **SQL** (Structured Query Language): **Standardisierte** (SQL1 1986 und SQL2 1992), **herstellerunabhängige** und **nicht-prozedurale** („Was ist das Problem?" statt „Wie löse ich das Problem?") **Programmiersprache** zum **Erzeugen** und **Betreiben** von relationalen Datenbanken.

- **DDL** (Data Definition Language): Befehle zum Erstellen (**CREATE**), Ändern (**ALTER**) oder Löschen (**DROP**) von **Datenbanken** oder darin enthaltenen Strukturen.

- **DML** (Data Manipulation Language): Befehle zum Einfügen (**INSERT**), Ändern (**UPDATE**), Löschen (**DELETE**) oder Abfragen (**SELECT**) von **Datenmengen**.

- **DCL** (Data Control Language): Befehle zur Wahrung der **Konsistenz** (**COMMIT**, **ROLLBACK**) und **Sicherheit** (**GRANT**, **REVOKE**) von Datenbanken.

- SQL: Statement in Data Definition Language

```
CREATE TABLE exemplar
(
  ex_isbn     CHAR(10)   NOT NULL,
  ex_lfd_nr   NUMBER(2)  NOT NULL,
  ex_status   CHAR(1)    NOT NULL,
  ex_zustand VARCHAR(1000),

  CONSTRAINT ex_primaerschluessel
    PRIMARY KEY (ex_isbn , ex_lfd_nr),

  CONSTRAINT ex_fremdschluessel_von_buch
    FOREIGN KEY (ex_isbn) REFERENCES buch (bu_isbn),

  CONSTRAINT ex_werte_fuer_status
    CHECK (ex_status IN ('P' , 'E' , 'R' , 'U'))
)
;
```

- SQL: Statement in Data Manipulation Language

- **Abfrage:**

```
SELECT   ex_isbn , ex_lfd_nr , bu_autor , bu_titel
FROM     exemplar , buch
WHERE    exemplar.ex_isbn   = buch.bu_isbn
AND      exemplar.ex_status = 'U'
ORDER BY bu_autor , bu_titel
;
```

- **Ergebnis** der Abfrage ist wieder eine **Tabelle**:

```
ex_isbn    ex_lfd_nr bu_autor     bu_titel
----------------------------------------------------
3150064619 01        Kant         Kritik der rein...
3150071143 03        Nietzsche    Jenseits von Gu...
3518100122 07        Wittgenstein Tractatus logic...
```

- SQL: Transaktionslogik

- **Transaktion**: Logisch zusammengehörige **Befehlsfolge**, die nur entweder **vollständig** oder **gar nicht** ausgeführt wird. Bsp. Geldüberweisung, Entlehnung.

- Transaktion wird von einem Benutzer der DB **gestartet** bzw. **beendet**, andere Benutzer sehen Datenänderungen erst **nach erfolgreichem Beenden** einer Transaktion. → Konsistenz der Daten bleibt auch bei Stromausfall während der Transaktion gegeben!

- **COMMIT: Transaktion beenden**, Aktualisierungen endgültig im System **speichern**.

- **ROLLBACK:** Ursprünglichen Zustand **vor der laufenden Transaktion** wieder herstellen.

- → **Sauer**, Hermann: „**Relationale Datenbanken** / Theorie und Praxis".

Der Autor, Ing. Mag. Michael Horvath, kann via selbstbefreiung@gmx.net kontaktiert werden.